IN THE CLAIMS

Claims 1, 4, 6-12, 14, 16-19, 21-25, 27, 28, 30-32, 34, 36 and 37 were pending at the time of the Office Action.

Claims 1, 4, 6-12, 14, 16-19, 21-25, 27, 28, 30-32, 34, 36 and 37 are now pending.

Claims 1, 4, 6-9, 11, 12, 17-19, 21, 22, 24, 25, 27, 28, 30-32 and 37 are currently amended.

Claims 2, 3, 5, 13, 15, 20, 26, 29, 33, and 35 were previously canceled**.**

Claims 1, 11, 12, 18, 19, 24, 25, and 32 are independent

Claim 38 is new.

1. (**Currently Amended**) A method comprising:

collecting initial entropy data, wherein the initial entropy data includes central processing unit data and operating system data, wherein the central processing unit data comprises:

(i) a timestamp counter;

(ii) a number of cache misses per second;

(iii) a number of branch mispredictions per second;

(iv) power fluctuations;

(v) a clock speed at which a central processing unit (CPU) is running; or

(vi) CPU-specific counters;

storing the initial entropy data in a nonvolatile memory;

updating the initial entropy data stored in the nonvolatile memory with newly collected entropy data; and

generating a string of random bits from the updated entropy data stored in the nonvolatile memory.

2. **(Canceled)**

3. **(Canceled)**

4. **(Currently Amended)** A method as recited in claim 1 wherein the initial entropy data includes operating system state information.

5. **(Canceled)**

6. **(Currently Amended)** A method as recited in claim 1 wherein the initial entropy data is maintained in a protected portion of an operating system kernel.

7. **(Currently Amended)** A method as recited in claim 1 wherein the method is executing on a system and the initial entropy data is inaccessible by an application program executing on the system.

8. (**Currently Amended**) A method as recited in claim 1 wherein generating a string of random bits includes hashing the updated entropy data to generate random seed data. ~~updating the entropy data includes hashing the entropy data stored in the nonvolatile memory with the newly collected entropy data~~.

9. (**Currently Amended**) A method as recited in claim 1 wherein updating the initial entropy data stored in the nonvolatile memory includes collecting new entropy data at periodic intervals.

10. (**Original**) A method as recited in claim 1 further including communicating the string of random bits to an application program requesting a random number.

11. (**Currently Amended**) One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:

collect initial entropy data, wherein the initial entropy data includes central processor- unit data and operating system data;

store the initial entropy data in a nonvolatile memory;

update the initial entropy data stored in the nonvolatile memory with newly collected entropy data; and generate a string of random bits from the updated entropy data stored in the nonvolatile memory.

12. **(Currently Amended)** A method comprising:

receiving a request for a random number;

retrieving, from a protected portion of an operating system kernel, initial entropy data that is regularly updated with newly collected entropy data, wherein the initial entropy data includes central processing unit data and operating system data;

hashing the updated entropy data to create random seed data;

generating a string of random bits from the random seed data; and

communicating the string of random bits to the requester of the random number.

13. **(Canceled)**

14. **(Previously Presented)** A method as recited in claim 12 wherein the central processing unit data includes data related to a state of a processor in a computer system and operating system data includes the state of an operating system executing on the computer system.

15. **(Canceled)**

16. (**Original**) A method as recited in claim 12 wherein the random seed data is maintained in a protected portion of an operating system kernel.

17. (**Currently Amended**) A method as recited in claim 12 wherein the updated entropy data is inaccessible by the requester of the random number.

18. (**Currently Amended**) One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:

receive a request for a random number;

retrieve initial entropy data from a protected portion of an operating system kernel, wherein the entropy data includes central processing unit data and operating system data;

hash the initial entropy data to create random seed data;

generate a string of random bits from the random seed data; and

communicate the string of random bits to the requester of the random number.

19 (**Currently Amended**) A method comprising:

collecting initial entropy data, wherein the initial entropy data includes central processing unit data and operating system data;

storing the initial entropy data in a protected portion of an operating system kernel; and generating a string of random bits based on the initial entropy data.

20. **(Cancelled)**

21. **(Currently Amended)** A method as recited in claim 19 wherein the initial entropy data is inaccessible by an application program.

22. **(Currently Amended)** A method as recited in claim 19 further comprising updating the initial entropy data with newly collected entropy data ~~by hashing entropy data in the protected portion of the operating system kernel with the newly collected entropy data~~.

23. **(Original)** A method as recited in claim 19 further comprising communicating the string of random bits to an application program requesting a random number.

24. **(Currently Amended)** One or more computer-readable memories containing a computer program that is executable by one or more processors, the computer program causing the one or more processors to:

collect initial entropy data from a central processing unit and an operating system executed by the central processing unit;

store the initial entropy data in a protected portion of an operating system kernel;

generate a string of random bits based on the initial entropy data.

25. **(Currently Amended)** An apparatus comprising:

a nonvolatile memory configured to store initial entropy data, wherein the initial entropy data stored in the nonvolatile memory is updated regularly with newly collected entropy data ~~by hashing the entropy data stored in the nonvolatile memory with newly collected entropy data~~; and

a random number generator, coupled to the nonvolatile memory, ~~to generate strings of random bits using the entropy data received from the nonvolatile memory~~, wherein ~~the entropy data is collected from a central processing unit and an operating system executed by the central processing unit~~ the random number generator utilizes the updated entropy data stored in the nonvolatile memory to generate strings of random bits.

26. **(Canceled)**

27. **(Currently Amended)** An apparatus as recited in claim 25 wherein the initial entropy data is updated at periodic intervals.

28. **(Currently Amended)** An apparatus as recited in claim 25 wherein the updated entropy data is maintained in a protected portion of an operating system kernel such that the entropy data is inaccessible by an application program.

29. **(Canceled)**

30. **(Currently Amended)** An apparatus as recited in claim 25 wherein the random number generator hashes the updated entropy data to generate random seed data.

31. **(Currently Amended)** An apparatus as recited in claim 25 further including a timer coupled to the random number generator, the timer indicating when to update the updated entropy data stored in the nonvolatile memory device.

32. **(Currently Amended)** One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:

collect entropy data from the one or more processors and one or more operating systems executed by the one or more processors;

store the collected entropy data in a nonvolatile memory;

update the entropy data stored in the nonvolatile memory with newly collected entropy data; and

produce a string of random bits from the entropy data stored in the nonvolatile memory, wherein the entropy data from the one or more processors comprises:

(i) a timestamp counter;

(ii) a number of cache misses per second ;

(iii) a number of branch mispredictions per second;

(iv) power fluctuations;

(v) a clock speed at which a processor is running; or

(vi) one or more processors-specific counters.

33. (**Canceled**)

34. (**Original**) One or more computer-readable media as recited in claim 32 wherein the entropy data is maintained in a protected portion of an operating system kernel.

35. (**Canceled**)

36. (**Original**) One or more computer-readable media as recited in claim 32 wherein to produce a string of random bits from the entropy data, the one or more processors hash the entropy data to generate random seed data.

37. (**Currently Amended**) One or more computer-readable media as recited in claim 32 wherein the entropy data stored in the nonvolatile memory is updated with newly collected entropy data at periodic intervals ~~by hashing the entropy data stored in the nonvolatile memory with the newly collected entropy data~~.

38. (**New**) A method as recited in claim 1 wherein generating a string of random bits includes:

(i) producing a first result by hashing the updated entropy data with a first hashing algorithm;

(ii) producing a second result by hashing the updated entropy data with a second hashing algorithm that is different from the first hashing algorithm; and

(iii) concatenating the first result with the second result.